

# RANGES AND INDICES ÖZELLİĞİ C# 8.0

## ▼ Nedir?

- Veri kümeleri olan, dizi, koleksiyon gibi yapılar üzerinde şu ana kada yapabildiğimiz özellikleri daha spesifik, efektif bir şekilde gerçekleştirmemizi sağlayan, güzel bir özelliktir.

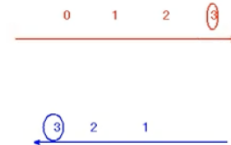
## Ranges and Indices


- C# 8.0 ile veri kaynakları üzerinde gerekli manipülasyonu sağlayabilmek ve bunun yanında kaynak içerisindeki tüm veriler üzerinde yapılan genel sorgulamalar ve algoritmalarından kaçınmak, yani direkt olarak hedef veri/ler odaklı çalışabilmek için yeni tipler ile operatörler geliştirilmiş bulunmaktadır.

•

## ▼ System.Index

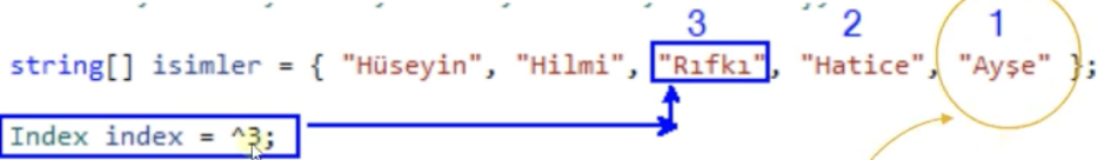
## System.Index



- Dizi ve koleksiyon yapılarındaki **index** kavramının tip olarak belirlenmiş halidir.
- Temelde index değerini bir tür ile tutmakla beraber  operatörüyle birlikte daha fazla anlam ifade etmekte ve dizinin index değerlerini tersine ifade edecek şekilde bir sorumluluk yüklenmektedir.



```
string[] isimler = { "Hüseyin", "Hilmi", "Rıfkı", "Hatice", "Ayşe" };  
Index index = 3;
```



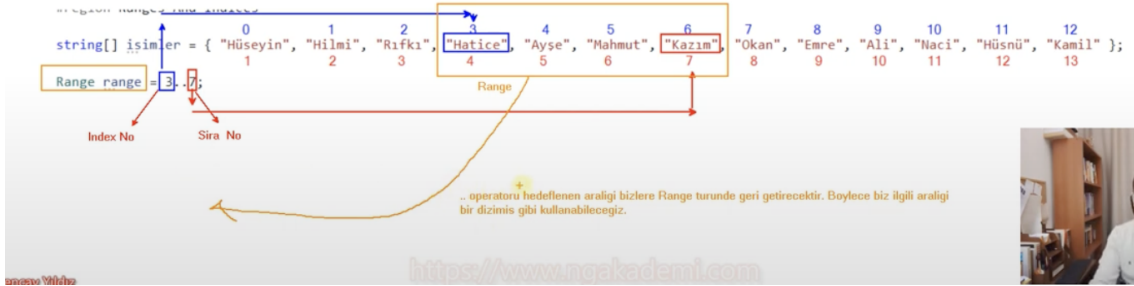
```
string[] isimler = { "Hüseyin", "Hilmi", "Rıfkı", "Hatice", "Ayşe" };  
Index index = ^3;
```

^ operatörü ile tersine index durumları kullanılırken index değerinin 0'dan değil 1'den başladığına dikkat ediniz.

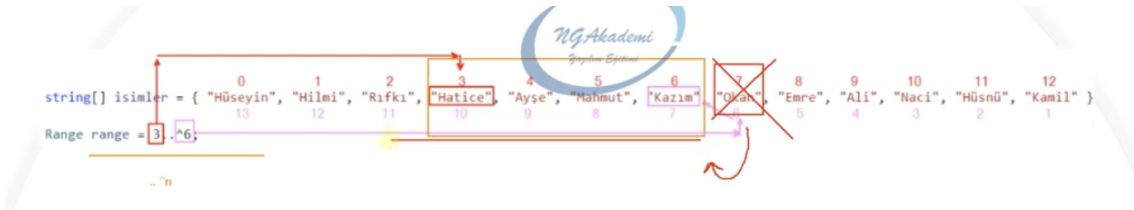
***Indexer[ ] operatörü içerisine tam sayı verilebildiği gibi Index türünde verilebilir.***

#### ▼ System.Range

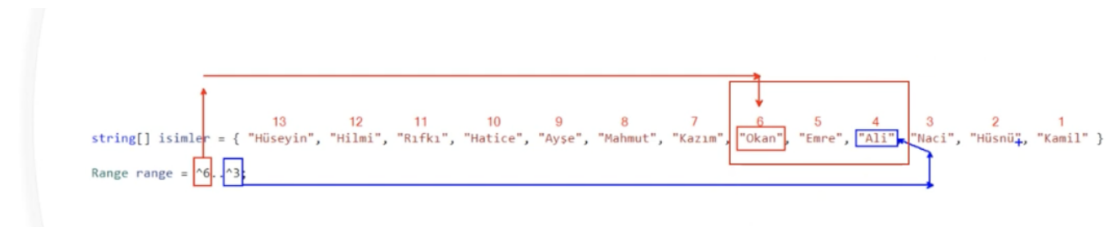
- İlgili dizi de hangi değer aralığında çalışacağımızı belirtmemize yarar
- Veri kümelerinde hangi değerler ile çalışacağımızı belirleyebilmek için index üzerinden aralık vermemizi ve bunu .. operatörü ile gerçekleştirmemizi sağlayan yapılanmadır.



- 3 .. ^n diyorsak, sağdan 1 ile başlayıp n. ye geldiğimizde n. yi allmayıp bir sonrakini alır. n+1. yi yani.



- Ama ^n .. 3 şeklinde kullanıyorsak, iki noktadan önce olduğundan n. alır.



**Indexer[] operatörü içerisine tam sayı veya Index türü verilebildiği gibi Range türünde verilebilir.**

```

U references
static void Main(string[] args)
{
    #region Ranges and Indices
    int[] sayilar = { 3, 5, 7, 9, 1, 2, 31, 321, 534, 5643, 457, 56, 78 };

    //Range range = 5..10;
    //var sayilar2 = sayilar[range];
    //sayilar2[0] = 100;

    Range range = ..; //--> Tüm diziye karşılık gelir.
    var sayilar2 = sayilar[range];
    sayilar2[0] = 100;

    #endregion
}

```

.. operatorunu sayısal değerler olmadan kullanırsanız ilgili diziyi topyekun kopyalar ve geriye döndürür.

```

U references
static void Main(string[] args)
{
    #region Ranges and Indices
    int[] sayilar = { 3, 5, 7, 9, 1, 2, 31, 321, 534, 5643, 457, 56, 78 };

    //Range range = 5..10;
    //var sayilar2 = sayilar[range];
    //sayilar2[0] = 100;

    //Range range = ..; //--> Tüm diziye karşılık gelir.
    //var sayilar2 = sayilar[range];
    //sayilar2[0] = 100;

    //Range range = 5..^3;
    //var sayilar2 = sayilar[range];

    Range range = ^8..^3;
    var sayilar2 = sayilar[range];

    #endregion
}

```

## ▼ .. operatörü

- Normalde index .. Sıra şeklinde veriyoruz fakat index .. index de verebiliriz.
- Veri kümelerinde belirliği bir aralığı temsil eden operatördür.
- Aralık operatörü diyede isimlendirilebilir.

Index     ..     Index

- .. operatörü sağına ve soluna sayısal bir değer alabildiği gibi özü itibariyle System.Index türünden de değerler alabilir.
- Geriye System.Range türünden yapı döndürür.

```

0 references
static void Main(string[] args)
{
    #region Ranges and Indices
    int[] sayilar = { 3, 5, 7, 9, 1, 2, 31, 321, 534, 5643, 457, 56, 78 };

    Index i1 = 5, i2 = 10;

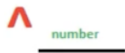
    Range range = i1..i2;
    var sayilar2 = sayilar[range];

    #endregion
}

```

## ▼ ^ operatörü

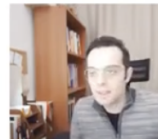
- Veri kümelerinde index değerinin tersini ifade eder.
- Normal index yapılanmasına nazaran ters index durumu 0'dan değil 1'den başlamaktadır.



- Genellikle bir dizinin son elemanına erişmek için kullandığımız karışık manevrasal algoritmalar bizleri kurtarmaktadır.

```
string isim = isimler[isimler.Length - 1];
```

```
string isim = isimler[^1];
```



Bu operatörü kullanabilmemiz için index türünde çalışmamız gerekmektedir. Çünkü bu operatör geriye index türü döndürür.